PoS Tagging Italiano Dantesco - Moderno

Federico Giacardi

Università di Torino

Abstract. Questo report riassume il lavoro svolto come progetto conclusivo per il corso di Tecnologie del Linguaggio Naturale, tenuto dal professor Mazzei presso l'Università di Torino nell'anno accademico 2024/2025.

Keywords: PoS Tagging · NLP · Viterbi Algorithm

1 Introduzione

L'obbiettivo del progetto è lo studio del problema del PoS tagging per l'Italiano moderno e per quello dantesco, cercando in particolare di implementare e confrontare i due principali approcci al task presentati durante il corso, ovvero HMM e CRF, per verificare l'effettiva differenza di prestazione e costo. Fatta questa premesse, il resto del lavoro è quindi organizzato come segue

1. Definizione dell'approccio utilizzato
2. Valutazione delle prestazioni

2 Approccio al problema

Come specificato dalla consegna, i dataset di training, validation, e testing sono quelli del progetto Universal Dependencies, disponibili a [6] e [5]. Il codice è stato scritto totalmente in Python, scelta che consente di beneficiare di un vasto ecosistema, utile soprattutto per l'implementazione del CRF, ma che penalizza parzialmente le prestazioni. Il formato .conllu dei treebank risulta piuttosto complesso, e contiene una serie di informazioni non necessarie al task in esame, per cui si è deciso di procedere con una fase di preprocessing, implementata nel modulo preprocessing.py, volta a trasformare il file nel seguente formato:
START
<parola 1> <PoS Tag 1>
...
<parola n> <PoS Tag n>
END
dove START ed END sono da interpretarsi come token speciali che denotano fine ed inizio della frase, consentendo di riconoscere la prima parola come tale, e di classificarla opportunamente. I file .conllu vanno messi nella directory raw_files, mentre il risultato viene salvato, con identico nome, ma in formato .txt, nella directory processed_files. I nomi dei dataset da utilizzare vanno specificati valorizzando le macro TRAINING_SET, TEST_SET, e DEV_SET presenti nel

modulo shared_data.py. I file così processati fanno da input per tutti i metodi di PoS tagging sperimentati, ovvero Majority Tagging, Hidden Markov Model e Conditional Random Field. L'ipotesi fondamentale che si intende validare con questo lavoro, che riassume quanto visto a teoria, è la seguente:

1. il Majority tagging è un approccio molto semplice, risultanto quindi altamente efficiente dal punto di vista computazionale, ma meno accurato
2. CRF è più efficiente, stante la possibilità di decidere sulla base di un insieme di features personalizzato per il problema, fatto che consente ad esempio un trattamento più preciso delle parole sconosciute. Per contro, risulta notevolmente più costoso dal punto di vista computazionale, dovendo apprendere i pesi delle features
3. HMM si pone come approccio intermedio, sia in termini di costo computazionale che di precisione

L'implementazione di questi approcci è data nei file majority.py, hmm.py e CRF_PoS_Tagging.ipynb. Majority tagging e HMM necessitano di due distribuzioni di probabilità: quella di transizione da tag a tag, e quella di emissione di una determinata parola dato il tag corrente. Le due distribuzioni in questione vengono stimate, come suggerito a teoria, tramite conteggio sul corpus, implementanto dal modulo training.py. Questa tecnica di stima, tuttavia, potrebbe portare a valori di probabilità nulli, per coppie di tag o di tag e parola non presenti nel corpus di training. Formalmente, vengono considerate sconosciute le parole non contenute in un vocabolario popolato a tempo di training. La compensazione di questo fenomeno richiede il ricorso a tecniche di smoothing, che consistono essenzialmente nella definizione di una distribuzione di probabilità della forma P(unk|tag). Le strategie testate nel progetto sono le seguenti

1. nouns_smoothing: prevede di assegnare sempre il tag NOUN alle parole sconosciute
2. nouns_verb_smoothing: prevede di assegnare sempre uno tra NOUN e VERB
3. uniform_smoothing: considera tutti i tag equiprobabili per la parola sconosciuta
4. single_word_smoothing: costruisce la distribuzione di probabilità usando i le frequenze dei token delle parole che occorrono una sola volta nel development set

Le stategie sono implementate da funzioni omonime presenti nel modulo smoothing.py, e possono essere selezionate valorizzando la costante SMOOTHING, definita nel modulo shared_data.py. Nell'implementazione del CRF, per fronteggiare il problema si è tratta ispirazione da quanto suggerito in [2], estendendo il feature template usato per rappresentare le singole parole con l'indicazione di prefisso (tre lettere iniziali), suffisso (tre lettere finali), e word form. Il resto del template è costruito ispirandosi sempre a [2], e prevede di considerare il contesto, definito in termini di parola precedente, con relativo pos tag, e successiva, oltre alla tipologia di parola (iniziale maiuscola e l'essere una cifra). L'implementazione del majority tagging e dell'algoritmo di Viterbi per il decoding di un HMM è data nel modulo decoding.py, mentre il CRF è implementato

nel notebook CRF_PoS_Tagging.ipynb. Il modulo evaluate.py funge infine da driver, occupandosi di richiamare i metodi della pipeline nell'ordine corretto e di valutare Majority Tagging ed HMM, tramite la stampa di classification report e matrice di confusione, calcolati con le funzioni di sklearn.

3 Valutazione dei risultati

Iniziamo valutando le prestazioni dell'algoritmo di Viterbi al variare della stregia di smoothing

Training Set	Development Set	Test Set	Accuracy	Smoothing
TrainOLD+VIT	DevOLD+VIT	TestOLD+VIT	88	single word
TrainVIT	DevVIT	TestVIT	88	single word
TrainOLD	DevOLD	TestOLD	88	single word
TrainOLD	DevOLD	TestVIT	73	single word
TrainVIT	DevVIT	TestOLD	70	single word
TrainOLD+VIT	DevOLD+VIT	TestOLD+VIT	86	nouns
TrainVIT	DevVIT	TestVIT	86	nouns
TrainOLD	DevOLD	TestOLD	85	nouns
TrainOLD	DevOLD	TestVIT	72	nouns
TrainVIT	DevVIT	TestOLD	65	nouns
TrainOLD+VIT	DevOLD+VIT	TestOLD+VIT	86	nouns-verbs
TrainVIT	DevVIT	TestVIT	86	nouns-verbs
TrainOLD	DevOLD	TestOLD	85	nouns-verbs
TrainOLD	DevOLD	TestVIT	72	nouns-verbs
TrainVIT	DevVIT	TestOLD	66	nouns-verbs
TrainOLD+VIT	DevOLD+VIT	TestOLD+VIT	88	uniform
TrainVIT	DevVIT	TestVIT	88	uniform
TrainOLD	DevOLD	TestOLD	87	uniform
TrainOLD	DevOLD	TestVIT	73	uniform
TrainVIT	DevVIT	TestOLD	70	uniform

I risultati ottenuti sono in linea con quanto atteso: smoothing uniforme e tramite conteggio su development set portano in effetti a risultati migliori rispettto all'assegnazione del solo tag NOUN o dei soli NOUN e VERB. Vale la pena osservare che le due strategie in questione sono praticamente equivalenti, anche se lo smoothing tramite conteggio su development set è più costoso dal punto di vista computazionale. Nel prosieguo del lavoro si utilizzerà lo smmothing basato su development set.

Procediamo poi riportando le prestazioni degli algorimi negli scenari richiesti dalla consegna

Possiamo osservare come la differenza di prestazioni sia effettivamente in linea con quanto atteso, e con lo stato dell'arte [4] [3] Il CRF si conferma superiore in tutte le situazioni, con un divario consistente sopratutto negli scenari cross domain, motivato dall'utilizzo di un ampio insieme di features per

Algoritmo	Training Set	Development Set	Test Set	Accuracy
Majority Tagging	TrainOLD+VIT	DevOLD+VIT	TestOLD+VIT	84
HMM	TrainOLD+VIT	DevOLD+VIT	TestOLD+VIT	88
CRF	TrainOLD+VIT	DevOLD+VIT	TestOLD+VIT	95
Majority Tagging	TrainVIT	DevVIT	TestVIT	83
HMM	TrainVIT	DevVIT	TestVIT	88
CRF	TrainVIT	DevVIT	TestVIT	95
Majority Tagging	TrainOLD	DevOLD	TestOLD	85
HMM	TrainOLD	DevOLD	TestOLD	88
CRF	TrainOLD	DevOLD	TestOLD	95
Majority Tagging	TrainOLD	DevOLD	TestVit	66
HMM	TrainOLD	DevOLD	TestVit	73
CRF	TrainOLD	DevOLD	TestVit	86
Majority Tagging	TrainVIT	DevVIT	TestOLD	69
HMM	TrainVIT	DevVIT	TestOLD	70
CRF	TrainVIT	DevVIT	TestOLD	77

la rappresentazione di parole sconosciute. Tuttavia, il costo computazionale di questo approccio è considerevole (il training dura circa 1 minuto sul corpus risultante dalla fusione dei due dataset). Il majority tagging si colloca, come teorizzato, sull'estremo opposto: è un algoritmo molto veloce ma con differenze di prestazione considerevoli, che vanno ad aumentare con il crescere della complessità dello scenario. I risultati ottenuti sono in linea con quelli indicati in [2]. L'HMM si conferma invece un'approccio intermedio, capace di accuracy notevole (anch'essa in linea con lo stato dell'arte [3]) e di un costo più contenuto rispetto a CRF. Le prestazioni particolarmente scadenti in situazioni cross-domain ci suggeriscono che la struttura sintattica dell'Italiano Moderno e di quello Dantesco differiscano notevolmente, cosa del tutto comprensibile. È interessante osservare che la differenza di prestazione tra l'HMM ed il Majoirty Tagging su questi dataset non è significativa dal punto di vista statistico (p-value = 0.5073). Questo stimola attente riflessioni sull'opportunità di utilizzo di un HMM, considerando che la differenza in termini di costo computazionale con il majority tagging è rilevante, mentre la diffrenza di prestazione no.

Risulta pertano ancor più interessante ed utile proseguire con un'analisi dettagliata degli errori commessi dall'algoritmo, servendosi in prima battuta dell'F1 Score, calcolato dal metodo classification_report di scikit-learn. Nel seguito, ci limitiamo a riportare le coppie classe-fscore più significative, un elenco completo dei dati è disponibile in allegato a questa relazione. Nel resoconto qui presentato si è proceduto a scartare le classi con supporto non significativo dal punto di vista statistico. Al netto degli scenari cross domain, in cui la prestazione risulta piuttosto scadente su tutte le classi, a riprova dei mutamenti nella satruttura dell'italiano, le classi più difficile da discernere sembrano essere aggettivi (1), verbi (4), nomi propri(5) e congiunzioni subordinate(14). Procediamo ora con l'analisi delle matrici di confusione dei diversi scenari, per identificare meglio la tipologia di errori commessi sulle classi in questione. Si noti che in alcuni casi vi

Training Set	Development Set	Test Set	F-Score	TAG
TrainVIT	DevVIT	TestVIT	0.73	ADJ
TrainVIT	DevVIT	TestVIT	0.77	VERB
TrainVIT	DevVIT	TestVIT	0.65	PROPN
TrainVIT	DevVIT	TestVIT	0.68	SCONJ
TrainOLD	DevOLD	TestOLD	0.70	ADJ
TrainOLD	DevOLD	TestOLD	0.51	PROPN
TrainOLD	DevOLD	TestOLD	0.70	SCONJ
TrainVIT-OLD	DevVIT-OLD	TestVIT-OLD	0.71	ADJ
TrainVIT-OLD	DevVIT-OLD	TestVIT-OLD	0.79	VERB
TrainVIT-OLD	DevVIT-OLD	TestVIT-OLD	0.65	PROPN
TrainVIT-OLD	DevVIT-OLD	TestVIT-OLD	0.79	NUM
TrainVIT-OLD	DevVIT-OLD	TestVIT-OLD	0.67	SCONJ
TrainOLD	DevOLD	TestVit	0.29	ADJ
TrainOLD	DevOLD	TestVit	0.68	ADV
TrainOLD	DevOLD	TestVit	0.75	NOUN
TrainOLD	DevOLD	TestVit	0.47	VERB
TrainOLD	DevOLD	TestVit	0.11	PROPN
TrainOLD	DevOLD	TestVit	0.70	AUX
TrainOLD	DevOLD	TestVit	0.16	NUM
TrainOLD	DevOLD	TestVit	0.61	PRON
TrainOLD	DevOLD	TestVit	0.61	SCONJ
TrainVIT	DevVIT	TestOLD	0.40	ADJ
TrainVIT	DevVIT	TestOLD	0.67	ADV
TrainVIT	DevVIT	TestOLD	0.66	NOUN
TrainVIT	DevVIT	TestOLD	0.53	VERB
TrainVIT	DevVIT	TestOLD	0.25	PROPN
TrainVIT	DevVIT	TestOLD	0.52	AUX
TrainVIT	DevVIT	TestOLD	0.75	DET
TrainVIT	DevVIT	TestOLD	0.72	PRON
TrainVIT	DevVIT	TestOLD	0.40	SCONJ

è un disallineamento tra gli indici della matrice di confusione e quelli del report, dovuti al fatto che nel secondo le classi mai incontrate vengono omesse.

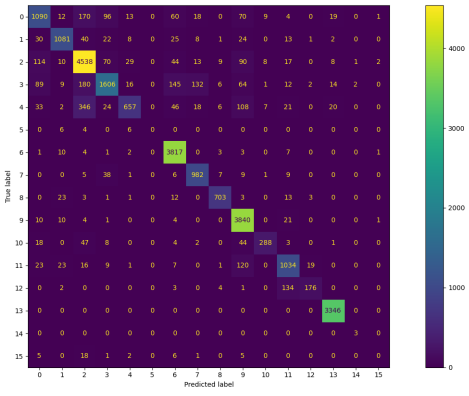

Fig. 1. Matrice di confusione per lo scenario TrainVIT, DevVIT, TestVIT
Gli aggettivi vengono confusi con sostantivi, verbi e determinativi. I verbi vengono confusi con aggettivi, sostantivi, apposizioni, e ausiliari. I nomi propri vengono confusi con sostantivi e determinativi. Le congiunzioni subordinate vengono confuse con i pronomi.

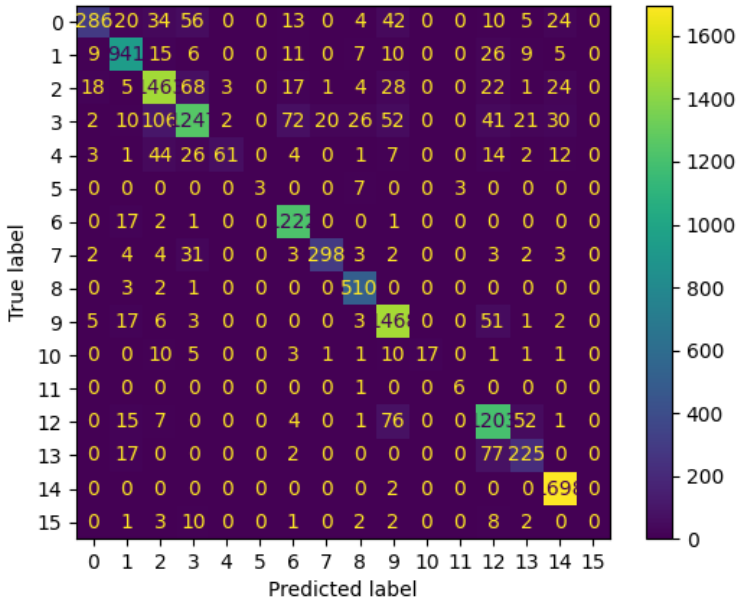

Fig. 2. Matrice di confusione per lo scenario TrainOLD, DevOLD, TestOLD

Gli aggettivi vengono confusi soprattutto con sostantivi, verbi e congiunzioni. I verbi vengono confusi soprattutto con sostantivi, apposizioni, e congiunzioni. I nomi propri vengono confusi soprattutto con sostantivi, e verbi. Le congiunzioni subordinate vengono confuse soprattutto con i pronomi.

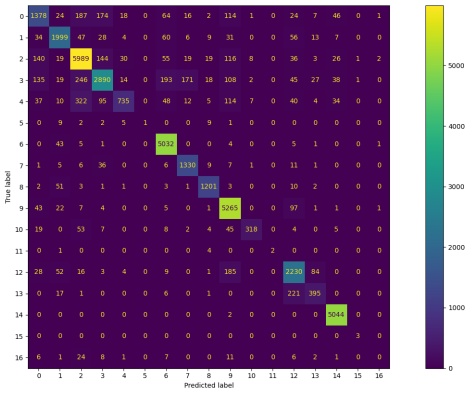

Fig. 3. Matrice di confusione per lo scenario TrainVIT-OLD, DevVIT-OLD, TestVIT-OLD

Gli aggettivi vengono confusi con sostantivi, verbi, e determinativi. I verbi vengono confusi con aggettivi, sostantivi, apposizioni, ausiliari, determinativi. I nomi propri vengono confusi con sostantivi, nomi propri e determinativi. Le congiunzioni subordinate vengono confuse con i pronomi.

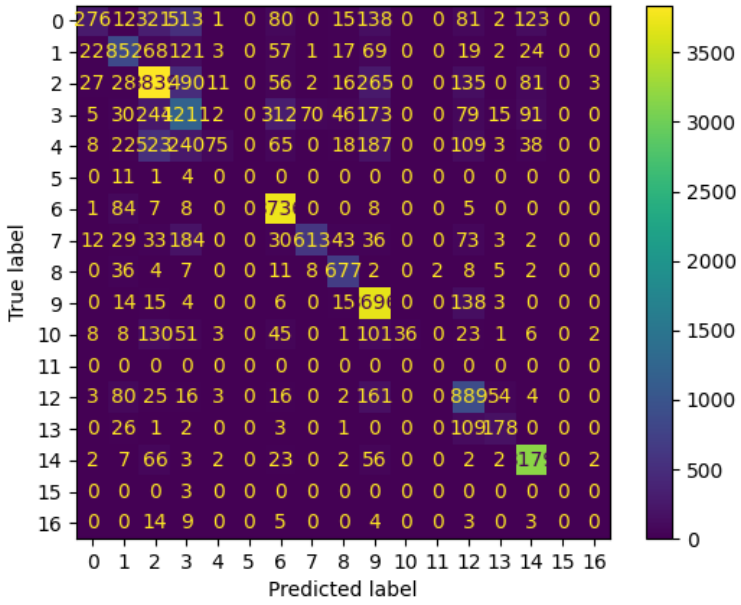

Fig. 4. Matrice di confusione per lo scenario TrainOLD, DevOLD, TestVIT
Gli aggettivi vengono confusi con sostantivi, verbi, determinativi e punteggiatura. I verbi vengono confusi con sostantivi, apposizioni, determinativi e pronomi I nomi propri vengono confusi con sostantivi, verbi, determinativi e pronomi Le congiunzioni subordinate vengono confuse con i pronomi.

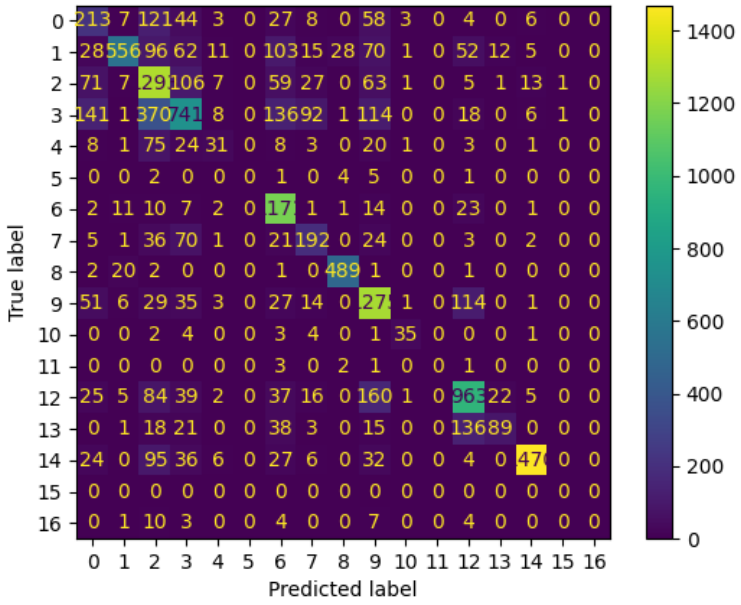

Fig. 5. Matrice di confusione per lo scenario TrainVIT, DevVIT, TestOLD
Gli aggettivi vengono confusi con sostantivi, verbi e determinativi. I verbi vengono confusi con sostantivi, apposizioni, determinativi e pronomi. I nomi propri vengono confusi con i sostantivi. Le congiunzioni subordinate vengono confuse con i pronomi.

Provando a generalizzare, possiamo affermare che

1. gli aggettivi vengono confusi soprattutto con sostantivi, verbi e determinativi
2. i verbi vengono confusi sopratutto con sostantivi ed apposizioni
3. i nomi propri vengono confusi con sostantivi e determinativi
4. le congiunzioni vengono confuse con i pronomi

Ad eccezione della confusione tra congiunzioni e pronomi, gli errori commessi avvengono tra categorie simili e ad elevata ambiguità, risultando quindi abbastanza plausibili. In più, osserviamo che essi sono in linea con quanto evidenziato da altre analisi per HMM su corpus italiani [1].

References

1. https://ceur-ws.org/Vol-1749/paper_014.pdf.
2. James Martin Daniel Jurafsky. *Speech and Language Processing (2nd Edition), publisher = Prentice-Hall, Inc.* 2009.
3. https://www.evalita.it/wp-content/uploads/2021/11/POS_UNIPI_ILC.pdf.

4. https://stanfordnlp.github.io/stanza/performance.html.
5. https://universaldependencies.org/treebanks/it_old/index.html.
6. https://universaldependencies.org/treebanks/it_vit/index.html.